2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Multi-Robot Formation Control via Consensus-Based Sliding Mode and Obstacle-Aware Adaptive Scaling

Hsien-I Lin† and Yu-Xian Chen
Institute of Electrical and Control Engineering
National Yang Ming Chiao Tung University
Hsinchu, Taiwan
sofin@nycu.edu.tw†

***Abstract*— This paper proposes a consensus-based sliding mode controller (CSMC) for multi-robot formation control. The framework integrates Laplacian-based consensus with sliding-mode robustness and adaptive formation scaling to simultaneously achieve accurate formation tracking and high formation consistency, while ensuring flexibility in constrained environments. The approach is validated in NVIDIA Isaac Sim and real-world experiments with Mecanum-wheeled robots. Compared with conventional sliding mode control (SMC), CSMC achieves consistent improvements in formation consistency, tracking accuracy, and overall performance in both simulation and real-world experiments. When compared with flocking-based approaches, CSMC provides substantially improved tracking performance and achieves better overall performance under consistency-prioritized evaluation metrics. These results demonstrate the effectiveness of CSMC in achieving reliable formation tracking, consistent coordination, and adaptive formation scaling for multi-robot navigation.**

I. Introduction

Driven by AI and IoT, Industry 4.0 increases demands for automation and flexibility. While single-robot systems are limited in sensing, payload, and scalability, multi-robot cooperation improves adaptability through task distribution and information sharing. Formation control [1]–[3] enables coordinated motion in applications such as exploration and surveillance; however, maintaining stability and real-time adaptability in dynamic environments remains challenging.

Moreover, from an application perspective, formation control frameworks intended for real-world deployment require systematic experimental validation and quantitative evaluation of formation-level performance. However, unified metrics for formation consistency and comprehensive real-world studies remain limited, making fair comparison across methods challenging.

To address these issues, we propose a consensus-based sliding mode formation control framework that enforces formation consistency while maintaining robust leader tracking. A virtual leader provides global motion references, while LiDAR-based perception, adaptive formation scaling, and APF-based repulsive adjustment enable shape-preserving adaptation in constrained environments.

The proposed approach is validated through both simulation in NVIDIA Isaac Sim and real-world experiments with mobile robots. Quantitative metrics are used to evaluate tracking accuracy and formation consistency, demonstrating the effectiveness and practicality of the proposed framework.

II. Related Work

A. Leader–Follower and Sliding Mode Formation Control

Among existing strategies, the leader–follower architecture is widely adopted, where one or more leaders define the global trajectory and followers maintain relative distance and orientation [4], [5]. Sliding mode control (SMC) has been integrated to enhance robustness, with controllers designed using tracking errors and Lyapunov-based analysis [6], [7]. Variants addressing chattering have also been reported [8], [9]. However, most methods are limited to two-robot formations or trajectory tracking and fail to address formation-wide consensus among multiple robots.

B. Consensus-Based Multi-Robot Formation Control

Consensus-based control has been widely adopted to overcome the limitations of leader–follower structures. Using graph-theoretic models, robots synchronize their states through neighbor information. Extensions combining proportional tracking, high-frequency switching, and sliding mode control have been proposed to improve convergence and handle time-varying formations [10], [11]. Graph-based flocking approaches further maintain inter-agent distances during navigation [12].

C. Obstacle Avoidance and Formation Adaptation

To improve adaptability, artificial potential fields (APF) are commonly integrated into decentralized formation control for obstacle avoidance [13], [14]. However, APF-based methods may cause formation compression in cluttered environments. Shape-changing strategies, such as adaptive V-formations [15], and formation scaling have been proposed to mitigate this issue, though some approaches rely on complex Laplacian formulations or multiple leaders, limiting practical applicability [16]. More recent studies incorporate learning-based adaptation or planning-based consensus to address non-convex environments [17], [18], While these approaches improve adaptability, they often introduce additional learning overhead, prior map assumptions, or system complexity.

979-8-3315-8160-2/26/$31.00 ©2026 IEEE

III. Problem Formulation

The formation control problem is structured into three components. Inter-robot communication is modeled as a directed graph using a Laplacian matrix. Tracking error dynamics are derived in each follower's local frame based on the virtual leader's state. A consensus error vector is then formulated via Kronecker products to enforce leader tracking and formation consistency.

A. Graph Model of the Multi-robot System

In the communication topology, the n follower robots are represented as nodes in a directed graph $G = (V, E)$, where each node $v_i \in V$ corresponds to robot R_i, and each directed edge $(v_i, v_j) \in E$ denotes that robot R_i can receive information from robot R_j. The adjacency matrix of G is defined as $A = (a_{ij}) \in \mathbb{R}^{n\times n}$, where $\forall i, j \in \{1, 2, \ldots, n\}$, $a_{ii} = 0$, and $a_{ij} = 1$ if information flows from R_j to R_i; otherwise, $a_{ij} = 0$.

The corresponding Laplacian matrix is given by $L = D - A \in \mathbb{R}^{n\times n}$, where the in-degree matrix is defined as $D = \mathrm{diag}(d_1, d_2, \ldots, d_n)$ with $d_i = \sum_{j=1, j\neq i}^{n} a_{ij}$.

We further consider a multi-robot system composed of n follower robots and a virtual leader. The extended topology includes $n+1$ nodes and is denoted by $\overline{G}$, with its associated Laplacian matrix defined as $\overline{L} = \overline{D} - \overline{A}$.

To facilitate the proposed formation control strategy, the following assumptions are made:

Assumption 1. All follower robots are connected to the virtual leader through a communication network, resulting in a directed and connected graph $\overline{G}$ that contains a spanning tree rooted at the virtual leader.

Assumption 2. The motion of the virtual leader is independent of the follower robots and is determined by a predefined trajectory or motion planning strategy.

Assumption 3. Each follower can reliably receive the state information (e.g., position, velocity, orientation) of the virtual leader.

Under these assumptions, the Laplacian matrix $\overline{L} \in \mathbb{R}^{(n+1)\times(n+1)}$ can be structured as:

$$\overline{L} = \begin{bmatrix} \mathbf{0}_{1\times 1} & \mathbf{0}_{1\times n} \\ L_{lf} & L_{ff} \end{bmatrix} \tag{1}$$

where $L_{lf} = -\mathbf{1}_{n\times 1}$ represents the uniform connection from the virtual leader to all followers, and $L_{ff} = L - \mathrm{diag}(L_{lf})$ is the modified Laplacian matrix of the follower subgraph.

This structure guarantees that $\overline{L}$ has a single zero eigenvalue corresponding to the virtual leader, indicating that all nonzero eigenvalues have positive real parts [19]. Consequently, L_{ff} is nonsingular and all of its eigenvalues have positive real parts, ensuring convergence of consensus dynamics.

B. Ideal Velocity Estimation and Formation Error Modeling

The formation's overall motion is guided by the movement of a virtual leader. Each follower is assigned a desired relative position with respect to the virtual leader and tracks a projected target point to preserve both inter-agent distance and formation orientation, as illustrated in Fig. 1.

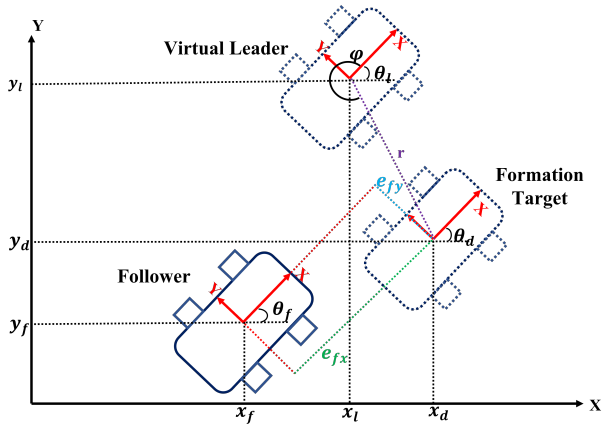


Fig. 1: Formation configuration

Given the virtual leader's pose $\mathbf{q}_l = \begin{bmatrix} x_l & y_l & \theta_l \end{bmatrix}^\top$, velocity $\mathbf{v}_l = \begin{bmatrix} v_{lx} & v_{ly} & \omega_l \end{bmatrix}^\top$, formation radius r, and angle offset φ, the ideal pose $\mathbf{q}_d$ for a follower is derived as:

$$\mathbf{q}_d = \mathbf{q}_l + r \begin{bmatrix} \cos(\theta_l + \varphi) \\ \sin(\theta_l + \varphi) \\ 0 \end{bmatrix}. \tag{2}$$

Differentiating Eq. (2), we obtain the time derivative of the desired pose:

$$\dot{\mathbf{q}}_d = \begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{\theta}_d \end{bmatrix} = \begin{bmatrix} \dot{x}_l \\ \dot{y}_l \\ \dot{\theta}_l \end{bmatrix} + r\omega_l \begin{bmatrix} -\sin(\theta_l + \varphi) \\ \cos(\theta_l + \varphi) \\ 0 \end{bmatrix}. \tag{3}$$

Assuming that the target point has the same orientation as the virtual leader (i.e., $\theta_d = \theta_l$), the desired velocity V_d in the leader's local frame is computed by:

$$\begin{aligned} \mathbf{v}_d = \begin{bmatrix} v_{dx} \\ v_{dy} \\ \omega_d \end{bmatrix} &= \begin{bmatrix} \cos(\theta_l) & \sin(\theta_l) & 0 \\ -\sin(\theta_l) & \cos(\theta_l) & 0 \\ 0 & 0 & 1 \end{bmatrix} \dot{\mathbf{q}}_d \\ &= \mathbf{v}_l + r\omega_l \begin{bmatrix} -\sin(\varphi) \\ \cos(\varphi) \\ 0 \end{bmatrix}. \end{aligned} \tag{4}$$

Given the ideal target pose and velocity, we define the tracking error between the follower's actual pose q_f and its assigned target q_d in the global frame as:

$$\mathbf{e} = \mathbf{q}_d - \mathbf{q}_f = \begin{bmatrix} x_d - x_f \\ y_d - y_f \\ \theta_d - \theta_f \end{bmatrix} = \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix}. \tag{5}$$

To express this error in the follower's local coordinate frame, the rotation matrix which transforms global vectors into the local frame is represented as :

$$Q(\theta_f) = \begin{bmatrix} \cos(\theta_f) & \sin(\theta_f) & 0 \\ -\sin(\theta_f) & \cos(\theta_f) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

and the formation tracking error expressed in the local coordinate frame is:

$$\mathbf{e}_f = Q(\theta_f)\mathbf{e} \tag{7}$$

taking its time derivative yields:

$$\dot{\mathbf{e}}_f = \dot{Q}(\theta_f)\mathbf{e} + Q(\theta_f)\dot{\mathbf{e}}. \tag{8}$$

Combining Eqs. (4)–(8), we derive the formation error dynamics in the follower's local frame as:

$$\begin{aligned}\dot{\mathbf{e}}_f &= \begin{bmatrix} 0 & \omega_f & 0 \\ -\omega_f & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{e}_f \\ &\quad + \begin{bmatrix} \cos(e_\theta) & -\sin(e_\theta) & 0 \\ \sin(e_\theta) & \cos(e_\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{v}_d - \mathbf{v}_f \\ &= \begin{bmatrix} 0 & \omega_f & 0 \\ -\omega_f & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{e}_f + \mathbf{v}_{d'} - \mathbf{v}_f \end{aligned} \tag{9}$$

The local-frame error dynamics serves as the basis for controller design, describing the relative motion between each follower and its designated formation reference.

C. Problem Statement

Throughout the formation process, all follower robots exchange pose and velocity information with one another while simultaneously receiving the state of the virtual leader. This communication architecture facilitates coordinated motion and preserves formation consistency.

A consensus error vector is formulated by integrating the individual tracking errors of all followers with the formation topology's Laplacian matrix through the Kronecker product:

$$\begin{aligned}\mathbf{e}_c &= \begin{bmatrix} \mathbf{e}_{c_1} & \mathbf{e}_{c_2} & \cdots & \mathbf{e}_{c_N} \end{bmatrix}^\top \\ &= (L_{ff} \otimes I_3) \begin{bmatrix} \mathbf{e}_{f_1} & \mathbf{e}_{f_2} & \cdots & \mathbf{e}_{f_N} \end{bmatrix}^\top \end{aligned} \tag{10}$$

where $\mathbf{e}_{c_i} \in \mathbb{R}^3$ denotes the consensus error of the i-th follower, $\mathbf{e}_{f_i} \in \mathbb{R}^3$ is the local tracking error relative to its desired target, $L_{ff} \in \mathbb{R}^{N\times N}$ is the Laplacian matrix of the follower subgraph, I_3 is the 3×3 identity matrix, and $\otimes$ denotes the Kronecker product.

This paper aims to design a sliding mode controller based on the consensus error $\mathbf{e}_c$ to ensure both coordination and stability of the formation. Specifically, the control law is designed to drive all components of $\mathbf{e}_c$ within a predefined bound δ, such that:

$$\lim_{t\to\infty} \|\mathbf{e}_{c_i}(t)\| < \delta, \quad \forall i \in \{1,2,\ldots,N\}. \tag{11}$$

This condition ensures that the formation consistently maintains coordination and tracking performance during motion.

IV. Proposed Method

To overcome the inherent decoupling between tracking and coordination in conventional approaches, we propose a Consensus-based Sliding Mode Controller that combines the robustness of sliding mode control with a Laplacian-based consensus topology. Each follower utilizes local neighbor information to compensate for formation errors while tracking its assigned target, thereby ensuring both system stability and inter-agent coordination.

A. Consensus-based Sliding Mode Controller

The control law is based on the following sliding surface:

$$\boldsymbol{\sigma}(t) = \mathbf{C}\,\mathbf{e}_c(t) + \int_{t_0}^{t} \mathbf{e}_c(\tau)\,d\tau, \quad \boldsymbol{\sigma} \in \mathbb{R}^{3N\times 1} \tag{12}$$

where $\mathbf{e}_c(t)$ is the consensus error vector of the entire system. The gain matrix $\mathbf{C} \in \mathbb{R}^{3N\times 3N}$ is a block-diagonal matrix composed of N submatrices $\mathbf{C}_i = \mathrm{diag}(c_x, c_y, c_\theta)$, each corresponding to the feedback strength for follower i in x, y, and θ directions. This sliding surface design enhances error response speed and improves steady-state convergence via the integral term.

To derive the control input, let $\boldsymbol{\sigma}_i = [\sigma_{xi}, \sigma_{yi}, \sigma_{\theta i}]^\top$ denote the sliding surface of follower i. By enforcing the ideal sliding condition $\dot{\boldsymbol{\sigma}}_i = \mathbf{0}$ and incorporating the error dynamics, the equivalent control input $\mathbf{u}_i^{eq}$ can be formulated as:

$$\mathbf{u}_i^{eq} = \begin{bmatrix} u_{xi}^{eq} \\ u_{yi}^{eq} \\ u_{\theta i}^{eq} \end{bmatrix} \tag{13}$$

where

$$\begin{aligned} u_{xi}^{eq} &= \omega_{f_i} e_{f_{y_i}} + v_{x_i}^{d'} + \frac{1}{c_x} e_{f_{x_i}} \\ &\quad - \frac{1}{D_i} \sum_{j\in N_i} \left(\omega_{f_j} e_{f_{y_j}} + v_{x_j}^{d'} + \frac{1}{c_x} e_{f_{x_j}} - u_{x_j} \right), \end{aligned} \tag{14}$$

$$\begin{aligned} u_{yi}^{eq} &= -\omega_{f_i} e_{f_{x_i}} + v_{y_i}^{d'} + \frac{1}{c_y} e_{f_{y_i}} \\ &\quad - \frac{1}{D_i} \sum_{j\in N_i} \left(-\omega_{f_j} e_{f_{x_j}} + v_{y_j}^{d'} + \frac{1}{c_y} e_{f_{y_j}} - u_{y_j} \right), \end{aligned} \tag{15}$$

$$\begin{aligned} u_{\theta i}^{eq} &= \omega_l + \frac{1}{c_\theta} e_{f_{\theta_i}} \\ &\quad - \frac{1}{D_i} \sum_{j\in N_i} \left(\omega_l + \frac{1}{c_\theta} e_{f_{\theta_j}} - u_{\theta_j} \right). \end{aligned} \tag{16}$$

D_i denotes the in-degree of follower i in the overall topology $\overline{G}$, representing the total number of information sources from the virtual leader and neighbors. N_i represents the set of neighboring followers directly connected to follower i within the follower subgraph defined by L_{ff}. In the equivalent control term, D_i is used as a normalization factor, and N_i defines the sources of information exchange and consensus error calculation.

To further enhance robustness and convergence speed, this paper applies a typical switching control term in sliding mode

control. Each component of the sliding surface is controlled via nonlinear terms defined for each $k \in \{x, y, \theta\}$ as follows:

$$\dot{\sigma}_{ki} = -K \cdot \frac{\sigma_{ki}}{|\sigma_{ki}| + \delta} \tag{17}$$

$$u_{ki}^{sw} = \frac{K}{D_i} \cdot \frac{\sigma_{ki}}{|\sigma_{ki}| + \delta} \tag{18}$$

where K is the switching gain, and δ denotes the boundary layer width to suppress chattering. This design ensures that the switching term quickly eliminates residual error while avoiding chattering near zero-crossings.

Combining the equivalent and switching terms, the final control output of follower i is:

$$\mathbf{u}_i = \mathbf{u}_i^{eq} + \mathbf{u}_i^{sw} \tag{19}$$

where $\mathbf{u}_i = [u_{xi}, u_{yi}, u_{\theta i}]^\top$ corresponds to the linear and angular velocity outputs of follower i.

In sliding mode control, the inclusion of integral terms can enhance error compensation and improve steady-state accuracy. However, if not properly designed, such terms may introduce instability or violate Lyapunov stability conditions. To overcome these challenges, we propose a bounded integration mechanism that ensures theoretical stability while maintaining practical implementability.

The integral term in Eq. (12) is redefined as:

$$\int \mathbf{e}_c(t)\, dt = \begin{cases} 0, & \text{if } |\mathbf{e}_c(t)| < \varepsilon \\ \int_{t_0}^{t} \mathbf{e}_c(\tau)\, d\tau, & \varepsilon \le |\mathbf{e}_c(t)| \text{ and } \\ & |\int \mathbf{e}_c(t)\, dt| < K \\ K, & \text{if } \int \mathbf{e}_c(t)\, dt > K \\ -K, & \text{if } \int \mathbf{e}_c(t)\, dt < -K \end{cases} \tag{20}$$

where, ε denotes the dead-zone threshold, and K represents the saturation bound. The design incorporates both dead-zone and saturation to handle two types of instability: when the error is small, the integral is suppressed to avoid direction mismatch between integral and error; when the integral exceeds the bound, it is clamped to prevent wind-up. This mechanism keeps the integral term aligned with the real-time error, ensuring the Lyapunov derivative remains negative and improving smoothness and stability under dynamic conditions.

B. Lyapunov-based Stability Analysis

To verify the stability of the proposed controller, this section first establishes a Lyapunov function for an individual follower i, and subsequently extends the analysis to the entire formation system.

We define the sliding surface Lyapunov function of follower i:

$$V_{\sigma_i} = \frac{1}{2} \sigma_i^\top \sigma_i > 0 \tag{21}$$

and its derivative:

$$\dot{V}_{\sigma_i} = \sigma_i^\top \dot{\sigma}_i = -K \sum_{k \in \{x,y,\theta\}} \frac{\sigma_{ki}^2}{|\sigma_{ki}| + \delta} \tag{22}$$

which shows that σ_i converges monotonically to zero as long as $K > 0$.

For error convergence, we define the energy function:

$$V_{e_i} = V_{xi} + V_{yi} + V_{\theta i} = \frac{1}{2} \left(e_{c_{xi}}^2 + e_{c_{yi}}^2 + e_{c_{\theta i}}^2 \right) \tag{23}$$

and its derivative:

$$\dot{V}_{e_i} = e_{c_{xi}} \dot{e}_{c_{xi}} + e_{c_{yi}} \dot{e}_{c_{yi}} + e_{c_{\theta i}} \dot{e}_{c_{\theta i}}. \tag{24}$$

Based on Eqs. (9), (10), (14)–(16), (18), and (19), each component can be expressed uniformly for each $k \in \{x, y, \theta\}$ as follows:

$$\dot{V}_{ki} = -K \frac{\sigma(e_{c_{ki}}) e_{c_{ki}}}{|\sigma(e_{c_{ki}})| + \delta} - \frac{1}{c_k} e_{c_{ki}}^2. \tag{25}$$

Given the integral term satisfies Eq. (20), we have:

$$e_{c_{ki}} \cdot \sigma(e_{c_{ki}}) = e_{c_{ki}}^2 + e_{c_{ki}} \int e_{c_{ki}}\, dt > 0. \tag{26}$$

Therefore, as long as $K > 0$ and $c_x, c_y, c_\theta > 0$, the Lyapunov derivatives associated with both the consensus error and the sliding surface remain negative definite.

To extend the stability analysis to the full multi-agent system, we define the global Lyapunov function as:

$$V_{\text{total}} = \sum_{i=1}^{N} (V_{\sigma_i} + V_{e_i}) > 0, \quad \dot{V}_{\text{total}} = \sum_{i=1}^{N} \left(\dot{V}_{\sigma_i} + \dot{V}_{e_i} \right) < 0. \tag{27}$$

In conclusion, the proposed consensus-based sliding mode controller ensures asymptotic stability at both the agent level and the overall system level.

C. Environment-aware Obstacle Avoidance Strategy

To enhance formation maneuverability in constrained environments, we propose a three-layer environment-aware obstacle avoidance strategy:

1. Global Environment Perception via LiDAR Fusion: Due to the virtual leader's lack of direct perception, we develop a sensing augmentation module that fuses LiDAR scans from multiple followers and transforms them into the virtual leader's frame. The process includes laser projection, frame transformation, point cloud merging and filtering, and virtual scan reconstruction.

First, each robot's `LaserScan` is projected into Cartesian coordinates and transformed into the virtual leader's frame. The merged point cloud is filtered to remove noise and outliers based on distance thresholds. To reduce computation, the fused data is converted back to a synthetic `LaserScan` by selecting the nearest point within each angular bin, enabling efficient navigation and obstacle processing, as shown in Fig. 2.

2. Obstacle-aware Adaptive Scaling: To avoid collisions in narrow passages, the formation scale is dynamically adjusted based on the synthesized scan. Let R denote the nominal formation radius—the maximum distance from the virtual leader to any follower under default spacing. For each angular direction θ, the available space is computed by

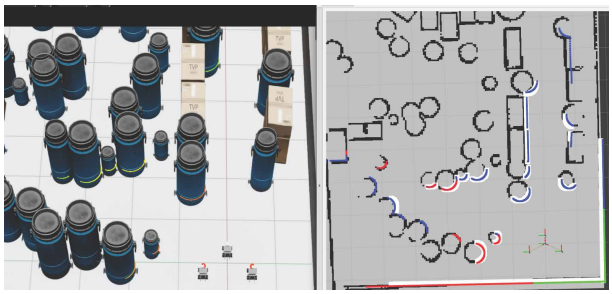


Fig. 2: LiDAR fusion results. Blue, green, and red denote the scans from Robot 1, 2, and 3, respectively, while white represents the fused global perception.

subtracting a safety margin d_{safety} from the obstacle distance $d(\theta)$:

$$s(\theta) = \frac{d(\theta) - d_{\text{safety}}}{R}, \quad s = \min_{\theta} s(\theta). \tag{28}$$

This yields the maximum allowable scaling factor s across all directions. To ensure stability, s is constrained within predefined bounds $s_{\min} \leq s \leq s_{\max}$ (e.g., $[0.6, 1.0]$) and smoothed using a first-order filter and rate limiter to prevent oscillations and abrupt changes.

3. Repulsive Adjustment via APF under Scaling Constraints: When scaling alone cannot resolve geometric constraints, we apply a repulsive adjustment using artificial potential fields (APF). Each physical follower activates a local APF controller when an obstacle is within d_{safe}, generating a repulsive force $\mathbf{f}_r \propto 1/d^2$. The force magnitude is modulated by the current scaling factor s—smaller s values reduce response strength to maintain formation stability.

The virtual leader also applies a simplified APF module. Based on the fused scan, a repulsive direction is estimated, and a small correction $\Delta\mathbf{v}$ is added to the navigation velocity $\mathbf{v}_{\text{nav}}$. This adjustment prevents wall-hugging and increases clearance for trailing robots. The correction is scaled by s to maintain a safety margin in wide spaces and minimize disruption in tight areas.

Together, these three layers provide robust, real-time adaptation to environmental constraints while preserving formation consistency and safety.

D. System Architecture

The proposed framework, illustrated in Fig. 3, is implemented within a ROS 2-based architecture to support modular design and real-time communication. It consists of a central console hosting the virtual leader module and multiple follower robots.

Each follower robot (Fig. 4) operates in a decentralized manner, handling local sensing and motion control. The virtual leader (Fig. 5) fuses LiDAR data from all followers to reconstruct the surrounding environment, performs navigation planning, and broadcasts desired states with adaptive formation scaling and Formation-level obstacle avoidance.

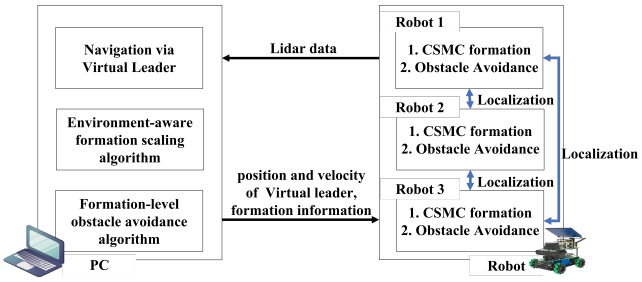


Fig. 3: System architecture

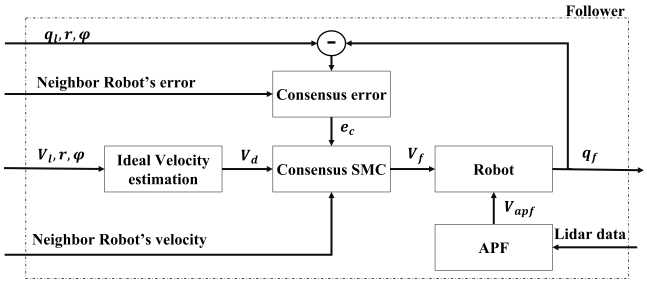


Fig. 4: System Architecture of Follower Robot

V. Experimental Results

A. Experimental Setup

Simulation experiments are conducted using NVIDIA Omniverse Isaac Sim 4.2 integrated with a ROS 2 framework. Real-world experiments are carried out using Mecanum-wheeled mobile robots equipped with 2D LiDAR and an IMU. High-level perception, navigation, and formation control modules are executed on an onboard NVIDIA Jetson Orin Nano 4GB platform, while low-level motor control is handled by an STM32-based embedded controller.

In terms of communication topology, as illustrated in Fig. 6, it is assumed that all followers can communicate with one another and have direct access to the state of the virtual leader. As a result, the overall network topology forms a directed spanning tree rooted at the virtual leader, ensuring effective information dissemination throughout the formation. Under this configuration, the communication topology of the formation system comprising one virtual leader and three physical followers can be described by the following

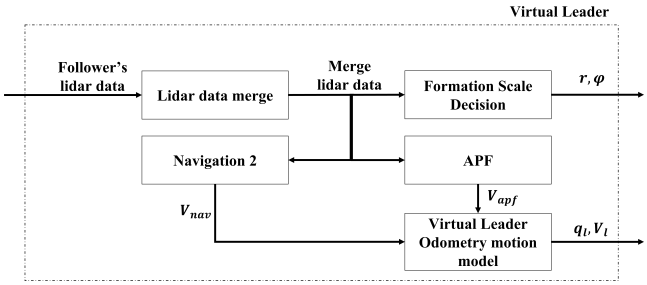


Fig. 5: System Architecture of the Virtual Leader

Laplacian matrix:

$$\overline{L} = \begin{bmatrix} \mathbf{0}_{1\times 1} & \mathbf{0}_{1\times n} \\ L_{lf} & L_{ff} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix} \tag{29}$$

where $\overline{L} \in \mathbb{R}^{4\times 4}$, the first row corresponds to the virtual leader node, and the remaining three rows describe the topological relationships among the physical followers. The submatrix L_{lf} indicates that each follower can receive state information from the virtual leader, while L_{ff} defines the adjacency and weight configuration among followers, which is used to construct the local consensus control term described in Section IV.

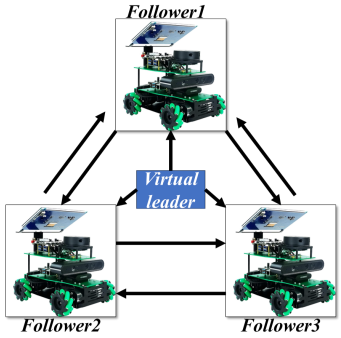
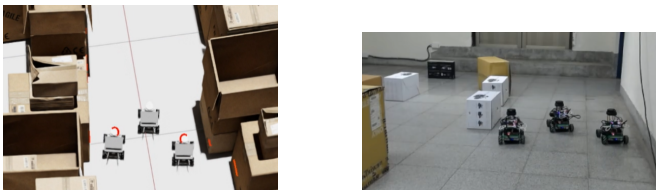


Fig. 6: Inter-Robot Communication Topology

To assess the effectiveness of the proposed method, the consensus-based sliding mode controller is compared with a conventional sliding mode controller [6] and a flocking-based formation control method [12]. The conventional SMC emphasizes leader-tracking accuracy but lacks explicit coordination among multiple robots, while the flocking-based approach focuses on maintaining formation consistency through neighbor interactions, often at the expense of tracking accuracy. These two methods represent typical tracking-oriented and consistency-oriented control strategies and therefore provide meaningful baselines for evaluation. All controllers are tested under identical experimental conditions.

The experimental platforms are illustrated in Fig. 7. Both simulation and real-world experiments are conducted under multiple representative scenarios. In simulation, each controller is executed seven times per scenario, while in real-world experiments, each controller is executed five times per scenario. Performance metrics are computed for each run and averaged for analysis.

One representative scenario involves narrow corridors (Fig. 8(a), Fig. 8(e), and Fig. 9(c)) to evaluate formation adaptability under spatial constraints. Another scenario considers a sparse–dense environment (Fig. 8(c), Fig. 9(a), and Fig. 9(e)), assessing the controller's ability to maintain tracking performance and formation integrity under varying obstacle densities.



(a) (b)

Fig. 7: Snapshots of the experimental platforms: (a) simulation environment; (b) real-world setup.

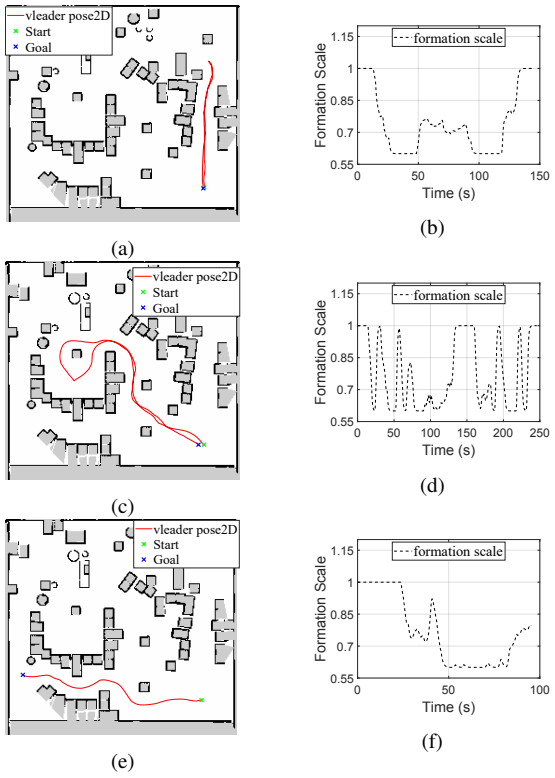


Fig. 8: Simulation results across multiple scenarios: (a), (c), (e) show the virtual leader trajectories with the map for Test 1–3, while (b), (d), (f) present the corresponding formation size variations.

B. Performance Metrics and Evaluation

To evaluate the effectiveness of the proposed method in both formation tracking and consistency, two categories of performance metrics are adopted:

- **Tracking Performance:** The mean absolute errors (MAE) of distance and orientation among the three robots:

$$\mathrm{MAE}_{\mathrm{dist}}(t) = \frac{1}{3}\left(|e_{d1}(t)| + |e_{d2}(t)| + |e_{d3}(t)|\right), \tag{30}$$

$$\mathrm{MAE}_{\mathrm{rad}}(t) = \frac{1}{3}\left(|e_{a1}(t)| + |e_{a2}(t)| + |e_{a3}(t)|\right) \tag{31}$$

  where $e_{di}(t)$ denotes the distance error and $e_{ai}(t)$ denotes the orientation error of robot i at time t.
- **Formation Consistency:** This is evaluated using the standard deviation of formation edge length ratios σ_{scale}

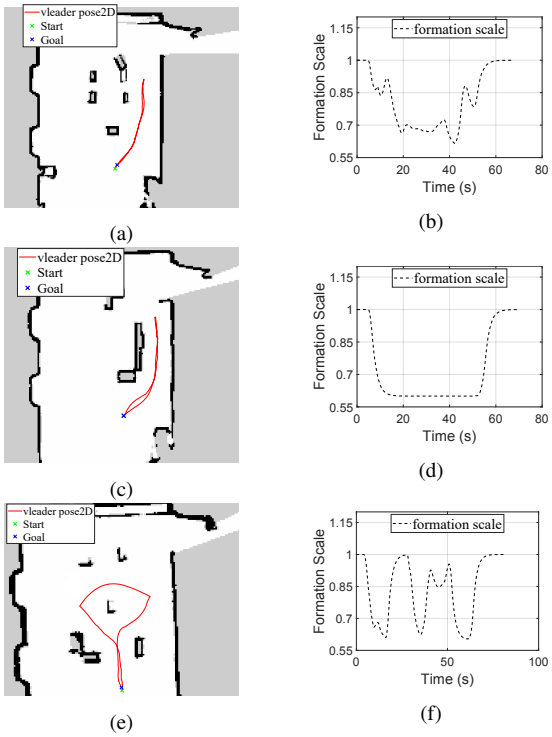


Fig. 9: Real-world experimental results across three scenarios: left column (a), (c), (e) shows the virtual leader trajectories with the map for Test 4–6, and right column (b), (d), (f) presents the corresponding formation size variations.

and a circular standard deviation $\sigma_{\text{circ}\theta}$ for robot orientation:

$$\sigma_{\text{scale}}(t) = \text{std}\left(\frac{d_{12}(t)}{t_{12}(t)}, \frac{d_{13}(t)}{t_{13}(t)}, \frac{d_{23}(t)}{t_{23}(t)}\right), \tag{32}$$

$$\bar{x}(t) = \frac{1}{n}\sum_{i=1}^{n}\cos\theta_i(t), \quad \bar{y}(t) = \frac{1}{n}\sum_{i=1}^{n}\sin\theta_i(t),$$

$$R(t) = \sqrt{\bar{x}^2(t) + \bar{y}^2(t)},$$

$$\sigma_{\text{circ}\theta}(t) = \sqrt{-2\ln R(t)} \tag{33}$$

where $\theta_i(t)$ is the heading angle of robot i at time t.

For each metric, the RMSE over the experiment duration is computed: $\text{RMSE}_{\text{dist}}$, RMSE_{rad}, $\text{RMSE}_{\sigma\text{scale}}$, and $\text{RMSE}_{\sigma\text{circ}\theta}$. These are combined into three weighted costs:

- **Tracking Cost:**

$$J_{\text{track}} = 0.8 \cdot \text{RMSE}_{\text{dist}} + 0.2 \cdot \text{RMSE}_{\text{rad}}. \tag{34}$$

This metric emphasizes position tracking as the primary objective, with angular error as a secondary factor.

- **Consistency Cost:**

$$J_{\text{cons}} = 0.8 \cdot \text{RMSE}_{\sigma_{\text{scale}}} + 0.2 \cdot \text{RMSE}_{\sigma_{\text{circ}\theta}}. \tag{35}$$

This emphasizes formation edge-length consistency as the primary evaluation criterion.

- **Overall Cost:**

$$J_{\text{overall}} = 0.7 \cdot J_{\text{cons}} + 0.3 \cdot J_{\text{track}}. \tag{36}$$

The overall cost prioritizes consistency while considering tracking performance as a secondary factor.

Using the aforementioned metrics and cost functions, controller performance is quantified in terms of tracking accuracy, formation consistency, and overall performance.

C. Results and Analysis

As summarized in Tables I and II, the proposed CSMC consistently outperforms the conventional SMC by improving formation consistency while maintaining or enhancing tracking accuracy, thereby achieving the best overall performance. In simulation, CSMC reduces consistency error, tracking error, and overall cost by average margins of 11%, 7%, and 10%, respectively. In real-world experiments, the improvements are 11%, 8%, and 10%. When compared with the Flocking approach, CSMC delivers substantial gains in tracking, with improvements of 69% in simulation and 49% in real-world tests. Although Flocking demonstrates slightly better consistency in certain cases, CSMC achieves 40% and 23% improvements in the overall performance metric, which prioritizes consistency.

TABLE I: Performance Comparison and Improvement Rates Across Scenarios (simulation)

Exp.	Ctrl.	J_{cons}	J_{track}	J_{overall}	% J_{cons}	% J_{track}	% J_{overall}
1	CSMC	0.0219	**0.0290**	**0.0240**	–	–	–
	SMC	0.0249	0.0316	0.0269	12.07	8.26	10.73
	Flocking	**0.0170**	0.0974	0.0411	-28.57	70.19	41.55
2	CSMC	0.0298	**0.0382**	**0.0323**	–	–	–
	SMC	0.0330	0.0400	0.0351	9.44	4.63	7.79
	Flocking	**0.0261**	0.1039	0.0494	-14.49	63.22	34.51
3	CSMC	0.0245	**0.0297**	**0.0261**	–	–	–
	SMC	0.0282	0.0323	0.0294	12.99	8.18	11.41
	Flocking	**0.0230**	0.1089	0.0487	-6.83	72.72	46.46
Avg.	CSMC	0.0254	**0.0323**	**0.0275**	–	–	–
	SMC	0.0287	0.0347	0.0305	11.37	6.84	9.82
	Flocking	**0.0220**	0.1034	0.0464	-15.46	68.75	40.77

VI. CONCLUSIONS

This work presented a consensus-based sliding mode formation control (CSMC) framework for multi-robot systems. The controller leverages relative error feedback from neighboring robots and the state of virtual leader to improve formation consistency and trajectory tracking. In addition, LiDAR fusion, adaptive formation scaling, and APF-based repulsive adjustment further enhance coordination in constrained environments. The proposed framework is validated in both NVIDIA Isaac Sim and real-world experiments with Mecanum-wheeled robots. In simulation, compared to conventional SMC, it achieves average improvements of 11% in formation consistency, 7% in tracking accuracy,

TABLE II: Performance Comparison and Improvement Rates Across Scenarios (Real-world)

Exp.	Ctrl.	J_{cons}	J_{track}	J_{overall}	% J_{cons}	% J_{track}	% J_{overall}
4	csmc	**0.0301**	**0.0294**	**0.0299**	–	–	–
	smc	0.0354	0.0348	0.0352	14.9	15.7	15.1
	flocking	0.0344	0.0627	0.0429	12.5	53.2	30.4
5	csmc	**0.0358**	**0.0311**	**0.0344**	–	–	–
	smc	0.0405	0.0344	0.0386	11.55	9.63	11.04
	flocking	0.0408	0.0639	0.0478	12.41	51.41	28.07
6	csmc	0.0383	0.0355	**0.0375**	–	–	–
	smc	0.0412	**0.0353**	0.0395	7.04	-0.65	4.98
	flocking	**0.0334**	0.0629	0.0423	-14.62	43.46	11.30
Avg.	csmc	**0.0347**	**0.0320**	**0.0339**	–	–	–
	smc	0.0390	0.0348	0.0378	10.98	8.18	10.20
	flocking	0.0362	0.0632	0.0442	4.13	49.36	23.48

and 10% in overall performance; in real-world experiments, the corresponding improvements are 11%, 8%, and 10%. Furthermore, when evaluated against the Flocking approach, CSMC yields substantially higher gains, with a 69% improvement in tracking performance in simulation and 49% in real-world experiments, accompanied by overall performance increases of 40% and 23%, respectively. These results confirm that the proposed framework achieves a favorable balance between formation tracking accuracy and formation consistency, enabling robust and adaptive multi-robot formation navigation in diverse and constrained scenarios. Future work will extend this framework to dynamic obstacle-rich environments, incorporate adaptive parameter tuning, and explore its deployment in large-scale, real-world multi-robot applications.

VII. Supplementary Material

A comprehensive demonstration video presenting experimental results across multiple environments and control methods in both simulation and real-world settings is provided.[1] An additional demonstration with five robots in simulation and real-world environments illustrates system scalability beyond the three-robot configuration used in the primary evaluation.[2] A formation switching demonstration with three robots is also provided to illustrate behavioral feasibility in reestablishing a stable formation after a shape transition.[3]

References

[1] Y. Liu, J. Liu, Z. He, Z. Li, Q. Zhang, and Z. Ding, "A survey of multi-agent systems on distributed formation control," *Unmanned Systems*, vol. 12, no. 5, pp. 913–926, 2024.

[2] L. Tsiu and E. D. Markus, "A survey of formation control for multiple mobile robotic systems," *International Journal of Mechanical Engineering and Robotics Research*, vol. 9, no. 11, pp. 1515–1520, 2020.

[3] M. M. Gulzar, S. T. H. Rizvi, M. Y. Javed, U. Munir, and H. Asif, "Multi-agent cooperative control consensus: A comparative review," *Electronics*, vol. 7, no. 2, p. 22, 2018.

[1] https://youtu.be/XYJc4wOJQYo

[2] https://youtu.be/uR9lIILNd00

[3] https://youtu.be/A3DzpuAVNSg

[4] J. Lagunas-Avila, R. Castro-Linares, and J. Alvarez-Gallegos, "Obstacle avoidance in leader-follower formation using artificial potential field algorithm," in *2021 18th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 2021, pp. 1–6.

[5] K. Tang, X. Pan, X. Guo, and C. Wu, "A cooperative control method of differential agv based on leader-follower strategy," in *2023 42nd Chinese Control Conference (CCC)*, 2023, pp. 3012–3016.

[6] F. Wu, J. Chen, and Y. Liang, "Leader-follower formation control for quadrotors," *IOP Conference Series: Materials Science and Engineering*, vol. 187, no. 1, p. 012016, 2017.

[7] J.-K. Lee, J.-B. Park, and Y.-H. Choi, "Tracking control of non-holonomic wheeled mobile robot based on new sliding surface with approach angle," *IFAC Proceedings Volumes*, vol. 46, no. 29, pp. 38–43, 2013, 3rd IFAC Symposium on Telematics Applications.

[8] B. Jiang, J. Li, and S. Yang, "An improved sliding mode approach for trajectory following control of nonholonomic mobile AGV," *Scientific Reports*, vol. 12, no. 1, p. 17763, 2022.

[9] H. Huang and J. Gao, "Backstepping and novel sliding mode trajectory tracking controller for wheeled mobile robots," *Mathematics*, vol. 12, no. 10, 2024.

[10] F. Wu, J. He, G. Zhou, H. Li, and Y. Liu, "Performance of sliding mode and consensus-based control approaches for quadrotor leader-follower formation flight," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 1671–1676.

[11] Q. Li, Y. Hua, X. Dong, and Z. Ren, "Time-varying formation tracking control for unmanned aerial vehicles: Theories and applications," *IFAC-PapersOnLine*, vol. 55, no. 3, pp. 49–54, 2022, 16th IFAC Symposium on Large Scale Complex Systems: Theory and Applications LSS 2022.

[12] D. Koung, I. Fantoni, O. Kermorgant, and L. Belouaer, "Consensus-based formation control and obstacle avoidance for nonholonomic multi-robot system," in *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2020, pp. 92–97.

[13] K. Dong, "Cooperative obstacle avoidance strategy of multi-robot leader formation based on artificial potential field method," *World Scientific Research Journal*, vol. 8, no. 2, pp. 520–527, 2022.

[14] Y. Zhang, J. Chen, M. Chen, C. Chen, Z. Zhang, and X. Deng, "Integrated the artificial potential field with the leader–follower approach for unmanned aerial vehicles cooperative obstacle avoidance," *Mathematics*, vol. 12, no. 7, 2024.

[15] D. N. Bui, M. D. Phung, and H. P. Duy, "Self-reconfigurable v-shape formation of multiple uavs in narrow space environments," in *2024 IEEE/SICE International Symposium on System Integration (SII)*, 2024, pp. 1006—-1011.

[16] X. Wu, R. Wu, Y. Zhang, and J. Peng, "Distributed formation control of multi-robot systems with path navigation via complex laplacian," *Entropy*, vol. 25, no. 11, 2023.

[17] C. Bai, P. Yan, W. Pan, and J. Guo, "Learning-based multi-robot formation control with obstacle avoidance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 11 811–11 822, 2022.

[18] L. E. Ruiz-Fernandez, J. Ruiz-Leon, D. Gomez-Gutierrez, and R. Murrieta-Cid, "Decentralized multi-robot formation control in environments with non-convex and dynamic obstacles based on path planning algorithms," *Intelligent Service Robotics*, vol. 18, no. 2, pp. 215–232, Mar 2025. [Online]. Available: https://doi.org/10.1007/s11370-024-00582-x

[19] H.-L. Wu, C.-C. Tsai, and F.-C. Tai, "Integral terminal sliding-mode formation control for uncertain heterogeneous networked mecanum-wheeled omnidirectional robots," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2018, pp. 1815–1820.